

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

A Media Agent

Inventors:

Wen-Yin Liu

Hong-Jiang Zhang

Zheng Chen

ATTORNEY'S DOCKET NO. MS1-933US

TECHNICAL FIELD

The following description relates to use of multimedia.

BACKGROUND

The number of images and other types of media content that are available to users via their computers, especially with the evolvement of the Internet, has become very large and is continuing to grow daily. For instance, people often download media content such as multimedia files, images, videos, audio, and so on from the World Wide Web (WWW). Additionally, a number of known computer programs simplify user generation of personalized media files. Moreover, multimedia files are often used to enhance documents and are typically distributed via e-mail as attachments.

It is very difficult to manage and utilize large and dynamic sets of media content or multimedia data (e.g., media from a web page, an email attachment, a multimedia generation tool, and so on.) once it is accessed or saved into a user's computing environment. For instance, once such data are saved into local folders, substantial numbers of accumulated multimedia files are typically never used again because they are difficult for the user to locate (e.g., through a search). This is often the case because media files themselves may be stored in an ad-hoc manner.

One conventional technique to facilitate a user's explicit search for media content requires the manual annotation of media content to identify semantics of the media. This technique is substantially limited for a number of reasons. One problem with this conventional technique to identify image semantics is that an image must be manually annotated prior to the user's actual search for media

1 content corresponding to the image. Another problem with this technique is that
2 manually annotating multimedia to include text is a tedious process that is prone to
3 human subjectivity and error. In other words, what one person may consider to be
4 semantically related (e.g., the subject matter, pertinent, interesting, significant, and
5 so on) to a particular image may be quite different from what another person may
6 consider to be semantically related to the particular image.

7 Another conventional technique to facilitate a user's explicit search for
8 media content analyzes text on a Web page text to identify semantics of images
9 displayed on the page. This analyzed text is compared to the user's search query.
10 If it matches to some extent, then the Web page may include media that is related
11 to the user's search. This technique is substantially limited in that images on a
12 Web page may have semantics other than what is specifically recited with the text
13 on the Web page.

14 The following arrangements and procedures address these and other
15 problems of managing and accessing multimedia data.

17 **SUMMARY**

18 The described arrangements and procedures provide a media agent to detect
19 and analyze inserted text . Based on the analysis, the media agent predicts or
20 anticipates whether a user intends to access media content. If so, the media agent
21 retrieves information corresponding to the anticipated media content from a media
22 content source. The media agent presents the retrieved media content based
23 information to the user as a suggestion.

24 . Responsive to user access of a media content source, the media agent
25 collects media content and associated text from the accessed media content source.

Semantic text features are extracted from the media content and the associated text. These semantic text features are indexed along the collected media content into a media database that may be personalized for the user.

The media agent monitors a user's actions to determine the user's media content use preferences. For instance, when the user's computer system is in an idle state (e.g., when the processor is not 100% active and has unused processing cycles), the agent collects media content and associated text from a media content source. Such an idle state may occur at any time, for instance, when a user is typing an e-mail message, and so on. The agent extracts semantic text features from the media content and the associated text. The agent determines that the media content is of interest to the user based at least in part on semantic similarity between the media content use preferences and the semantic text features. If the media agent determines that the media content is of interest to the user, the agent indexes the semantic text features into the user's personal media database.

BRIEF DESCRIPTION OF THE DRAWINGS

The same numbers are used throughout the drawings to reference like features and components.

Fig. 1 illustrates an exemplary environment in which the invention can be practiced.

Fig. 2 shows an exemplary host computer to semantically index, suggest, and retrieve media content according to personal usage patterns.

Fig. 3 shows exemplary aspects of process and data flows between modules and data sinks in the media agent module. Specifically, Fig. 3 shows sequences in

1 which data transfer, use, and transformation are performed during the execution of
2 the media agent.

3 Fig. 4 shows an exemplary procedure to automatically collect, manage, and
4 suggest information corresponding to personalized use of media content. More
5 specifically, Fig. 4 shows a procedure to determine whether offline gathering of
6 media content semantics, online gathering of media content semantics, preference
7 and intention modeling, or user intention prediction and suggestion procedures
8 should be performed.

9 Fig. 5 shows further aspects of an exemplary procedure to automatically
10 collect, manage, and suggest information corresponding to personalized use of
11 media content. More specifically, Fig. 5 shows further aspects of a procedure for
12 a media agent of Figs. 2 and 3 to perform online gathering of media content
13 semantics and preference and intention modeling.

14 Fig. 6 shows further aspects of exemplary procedures to automatically
15 collect, manage, and suggest information corresponding to personalized use of
16 media content. More specifically, Fig. 6 shows further aspects of a procedure for
17 a media agent of Figs. 2 and 3 to determine whether preference and intention
18 modeling or user intention prediction and suggestion procedures should be
19 performed.

20 Fig. 7 illustrates aspects of an exemplary suitable operating environment in
21 which a media agent to semantically index, suggest, and retrieve media content
22 according to personal usage patterns may be implemented.

23 Fig. 8 shows an exemplary user interface to present media suggestions
24 (e.g., filenames) for a user to select based on what the user has typed into a
25 window such as an e-mail message or other document.

DETAILED DESCRIPTION

The following description sets forth exemplary subject for a media agent to semantically index, suggest, and retrieve media content and other information corresponding to media content according to a user's personal media use patterns. The subject matter is described with specificity to meet statutory requirements. However, the description itself is not intended to limit the scope of this patent. Rather, the inventors have contemplated that the claimed subject matter might also be embodied in other ways, to include different elements or combinations of elements similar to the ones described in this document, in conjunction with other present or future technologies.

Overview

As discussed in the background section, using conventional techniques it is very difficult to manage and utilize large and dynamic sets of media content once it is accessed or saved into a user's computing environment because media files themselves may be stored in an ad-hoc manner. However, this is also the case because locating a particular multimedia file that is relevant to a context within which the user is working (i.e., the user's intent) is a substantially difficult problem. And, unless the user is performing an explicit search for media content, none of the described conventional procedures take into account multimedia content needs of the user within the context that he or she is working.

In contrast to such conventional procedures, the following arrangements and procedures provide for an intelligent media agent to autonomously collect semantic multimedia data text descriptions on behalf of a user whenever and wherever the user accesses multimedia data. The media agent analyzes these semantic multimedia data text descriptions in view of user behavior patterns and

actions to assist the user in identifying multimedia content that is appropriate to the context within which the user is operating or working. To accomplish this, the media agent provides timely prompts with suggested multimedia content and/or information corresponding to media content (e.g., suggested media filenames).

Fig. 1 illustrates an exemplary environment to identify a context within which the user or client is working and suggest semantically related multimedia content for the client to work with based on the identified context. In environment 100 one or more (x) clients 102 are coupled to a media content store 104. The media content store 104 is any combination of local storage (e.g., local volatile or non-volatile memory), networked storage (e.g., a parallel connection, an organizational intranet network, the Internet, and so on), or other communication configurations.

These communication configurations provide for electronic exchange of information using an appropriate protocol (e.g., TCP/IP, UDP, SOAP, etc.) between the host device 102 and one or more media content sources or servers that include multiple (y) pieces of media content 106. This electronic exchange provides for client 102 communication with media content store 104 to access (e.g., view, search, download, etc.) pieces of media content 106.

The storage of media content pieces 106 within media content store 104 can be arranged in any of a wide variety of manners and according to any of a wide variety of data formats. For example, media content pieces 106 may be stored on multiple servers hosting Web pages accessible via a network using an appropriate protocol such as Hypertext Transfer Protocol (HTTP). Web pages are documents that a user can view or otherwise render and which typically include links to one or more other pages that the user can access. Web pages are typically stored as

1 one or more files at a remote location(s), being accessed by the user via a
2 computer that is operatively coupled to a network. Web pages often include
3 multiple pieces of media content 106.

4 Media content pieces 106 include any of a wide variety of conventional
5 media content, such as audio content, video content (for example, still images or
6 frames of motion video), multimedia content, etc. A piece of media content 106
7 refers to media content that can be rendered, such as a single visual image, an
8 audio clip (e.g., a song or portion of a song), a multimedia clip (e.g., an
9 audio/video program or portion of an audio/video program), etc. The described
10 arrangements and procedures can be used with a wide variety of conventional
11 media content.

12 In the illustrated example, a user of a client 102 accesses the media content
13 store 104 for pieces of media content 106. The client 102 automatically detects a
14 user's access or utilization of a media object 106 (e.g., an image, a chart, an audio,
15 a video, an Excel ® file, etc.) and collects semantic text descriptions of the
16 accessed media object 106 during its use. These media object descriptions are
17 extracted from text associated with an accessed media content piece 106.

18 Media content 106 may coexist with corresponding text description. The
19 source of the text description may be part of the multimedia content itself or can
20 be independent but semantic to the content. For instance, an e-mail message may
21 describe attached media content (e.g., an attached image or video clip). Many
22 other composite documents, including slide files, word processor documents, and
23 so on, also commonly include both media content and corresponding text. All of
24 these files can be used as potential sources of semantic features of media data.
25 Thus, the client 102 collects or extracts semantic features of a media file from text

1 descriptions from the media content's environment (e.g., the Web page, the e-mail,
2 the compound document, and so on).

3 As a user operates within the computing environment of a client 102, the
4 client 102 monitors the user's activities and provides suggestions of semantically
5 related media content 106 to use based on these user activities in view of the
6 collected media object descriptions. For instance, after authoring a paragraph of
7 text description during preparation of a technical report, a user indicates that he or
8 she wants to insert some multimedia content 106 (e.g., a diagram). (There are any
9 numbers of ways for the user to author such a paragraph such as via user input
10 with a keyboard, a stylus, a mouse or other pointing device, voice recognition, and
11 so on). The client 102 anticipates the desire to insert the content and/or the
12 particular content that the user wishes to insert. This is accomplished by mapping
13 information from surrounding text description (e.g., text above, below, or to the
14 sides of the media content insertion point) to user prediction and preference
15 patterns and stored multimedia data text descriptions. Using this information, a
16 list of one or more anticipated multimedia items is presented (e.g., in a dialog box)
17 to the user for user selection. An example of this is described in greater detail
18 below in reference to Fig. 8.

19 **An Exemplary System**

20 Fig. 2 shows an exemplary host computer 102 to semantically index,
21 suggest, and retrieve media content according to personal usage patterns. The host
22 computer 102 is operational as any one of a number of different computing
23 devices such as a personal computer, an image server computer, a thin client, a
24 thick client, a hand-held or laptop device, a multiprocessor system, a
25 microprocessor-based system, a set top box, programmable consumer electronics,

1 a wireless phone, an application specific integrated circuit (ASIC), a network PC,
2 minicomputer, mainframe computer, and so on.

3 The host computer includes a processor 202 that is coupled to a system
4 memory 204. The system memory 204 includes any combination of volatile and
5 non-volatile computer-readable media for reading and writing. Volatile computer-
6 readable media includes, for example, random access memory (RAM). Non-
7 volatile computer-readable media includes, for example, read only memory
8 (ROM), magnetic media such as a hard-disk, an optical disk drive, a floppy
9 diskette, a flash memory card, a CD-ROM, and so on.

10 The processor 202 is configured to fetch and execute computer program
11 instructions from program modules 206; and configured to fetch data 208 while
12 executing the program modules 206. Program modules typically include routines,
13 programs, objects, components, data structures, etc., for performing particular
14 tasks or implementing particular abstract data types. For instance, program
15 modules 206 include the media agent module 210, and other applications (e.g., an
16 operating system, a Web browser application, and so on.

17 The media agent module 210 includes on-line crawler 212 and off-line
18 crawler 214 modules, a prediction module 216, a media search engine 218, a
19 suggestion module 220, and a self-learning module 222, each of which are
20 described in greater detail below. The media agent module 210 automatically
21 detects user actions with respect to media content to trigger one or more
22 appropriate modules 212 through 222. Media content (e.g., Web pages, composite
23 documents that include media content such as e-mails, word processing files, and
24 so on) refers to any one or more of the media content pieces 106 of Fig. 1 and/or
25 media represented in a user's personal media database 226.. Actions with respect

1 to media content include, for example: accessing a URL (e.g., with respect to a
2 media content piece 106), creating a media object, importing or downloading a
3 media object, inserting a media object (e.g., into a document), opening, saving,
4 updating or editing a media object, exporting or uploading a media object 106, and
5 so on.

6 The Online and Offline Media Content Crawler Components

7 The online 212 and offline 214 crawler modules are triggered at various
8 times to: (a) collect potentially related high-level features (also referred to herein
9 as semantic text features) of a media object from a composite document (e.g., an
10 e-mail, Web page, or word processing document); (b) extract semantic text
11 features from the media object itself; and (c) index the media object in the
12 personal media database 226 using the collected and extracted semantic text. A
13 composite document includes both media content and corresponding text (e.g., an
14 e-mail message with an attached picture or slide file, word processor documents,
15 etc.). For instance, if the composite document is an e-mail with an attachment, the
16 crawlers 212 and 214 may extract semantic text features (e.g., words) from both
17 the body of the e-mail message and from the attached media content piece itself.

18 Specific actions that trigger the on-line crawler module 212 include, for
19 example: visiting a URL, saving/downloading a media object from the Web or an
20 email, saving a media hyperlink from the Web, inserting a media object or its link
21 into a document or an e-mail, and so on. The off-line crawler module 214 is
22 activated at system 102 idle time to collect and index semantic text corresponding
23 to media objects local or remote to the host 102 (e.g., the media content pieces 106
24 of Fig. 1) that are similar to the user's preferences models 230. (User preferences
25 models 230 are described in greater detail below).

Media content semantic text features are extracted by crawlers 212 and 214 in a variety of different manners. For instance, text features are extracted based on up to six aspects of the text associated with media content: (1) a filename and identifier, (2) an annotation, (3) alternate text, (4) surrounding text, (5) a page title, and/or (6) other information. Note that all of these aspects may not be associated with each media content piece, and thus features are not extracted based on aspects that are not available for the media content piece.

(1) Image filename and identifier: each image is identified by a filename that is typically part of a larger identifier that indicates where the file is located (e.g., a URL). Often meaningful names are used as filenames and/or the identifier (e.g., URL) for an image. Each word in the filename and identifier can be used as a text feature. In one implementation, a set of rules is used to judge the usefulness of the filenames and URL for an image, and thereby limit the words used as text features.

One rule is that the filename be segmented into meaningful key words. Based on a standard dictionary (or alternatively a specialized dictionary), the filename is analyzed to determine whether it includes one or more words that are in the dictionary. Each such word is identified as a key word. For example, the filename "redflower.jpg" would be segmented into the key words "red" and "flower", each of which would be a text feature (assuming they each existed in the dictionary).

Another rule or criteria is that certain common words (e.g., articles) are excluded from being considered key words. For example, the filename "theredflower.jpg" could be segmented into the words "the", "red", and "flower", but only "red" and "flower" would be text features (the word "the" is a stop-word

1 and thus not identified as a key word). Other insignificant characters and groups
2 of characters can also be excluded, such as digits, hyphens, other punctuation
3 marks, filename extensions, and so on.

4 Another rule applies to the URL for an image. A URL typically represents
5 the hierarchy information of the image. The URL is parsed and segmented to
6 identify each word in the URL, and then resulting meaningful key words are used
7 as text features. For example, in the URL “. . . /images/animals/anim_birds.jpg”,
8 the words “animals” and “birds” are meaningful key words that would be
9 extracted as images. A dictionary can be used to identify the meaningful key
10 words as discussed above. For example, the word “images” would not be
11 meaningful as only images are being analyzed.

12 (2) Image annotation: each image can have a corresponding image
13 annotation which is a text label describing the semantics of the image, typically
14 input by the creator of the image file. This image annotation is intended to
15 describe the semantics of the image. Thus, each word in the image annotation
16 may be a key feature (although certain common words and/or insignificant
17 characters/character groups can be excluded as discussed above regarding image
18 filenames and identifiers).

19 (3) Alternate text: many web pages include alternate text for images. This
20 alternate text is to be displayed in place of the image in certain situations (e.g., for
21 text-based browsers). As this alternate text is intended to replace the image, it
22 often includes valuable information describing the image. Thus, each word in the
23 alternate text is a key feature (although certain common words and/or insignificant
24 characters/character groups may be excluded as discussed above regarding image
25 filenames and identifiers).

1 (4) Surrounding text: many web pages have text surrounding the images
2 on the rendered web page. This text frequently enhances the media content that
3 the web page designers are trying to present, and thus is frequently valuable
4 information describing the image. Thus, key words from the text surrounding the
5 image (e.g., text above the image, below the image, to the left of the image, and to
6 the right of the image) are extracted as text features (certain common words and/or
7 insignificant characters/character groups may be excluded as discussed above
8 regarding image filenames and identifiers). The amount of text surrounding an
9 image from which key words are extracted can vary. For instance, the three lines
10 (or sentences) of text that are closest to (adjacent to) the image are used, or
11 alternatively the entire paragraph closest to (adjacent to) the image can be used.
12 Alternatively, if information is available regarding the layout of the web page,
13 then the single sentence (or line) most related to the image can be used.

14 (5) Page title: many times a web page will have a title. If the web page
15 does have a title, then key words are identified in the title and used as text features
16 (certain common words and/or insignificant characters/character groups may be
17 excluded as discussed above regarding image filenames and identifiers).

18 (6) Other information: other information from the web page may also be
19 used to obtain words to be used as text features associated with an image. For
20 example, each URL on the page that is a link to another web page may be parsed
21 and segmented and meaningful key words extracted from the URL (analogous to
22 the discussion above regarding extracting meaningful key words from the URL of
23 the image). By way of another example, meaningful key words may be extracted
24 from "anchor text" that corresponds to the image. Anchor text refers to text that is
25 identified on the web page as text that should be kept near or next to the image

(e.g., which would cause the browser to move the text to a next page if the image were to be displayed on the next page). Key words can be extracted from the anchor text analogous to the discussion above regarding extracting meaningful key words from the alternate text.

After applying these various rules, the crawler 212 or 214 has a set of words that are text features extracted from the image. Note that certain words may be extracted multiple times and thus appear in the set multiple times. The crawler module 212 or 214 stores these high-level semantic text features and an identifier of the media content piece (e.g., a URL) in personal media content and features database 226. The media content piece itself may also optionally be stored in a separate database 236 from the high-level semantic text features.

The extracted high-level text features are a set of words. The crawler module 212 or 214 takes the extracted features for media content from personal media content database 226 and indexes the media content piece. These generated feature vectors or indices are stored in personal media content database 226 or alternatively elsewhere. The indexing process refers to generating, as necessary, feature vectors corresponding to the media content piece and storing a correlation between the generated feature vectors and the media content piece.

The crawler module 212 or 214 converts the extracted high-level text features into a text feature vector D_i for image i using a well-known TF*IDF method:

$$D_i = TF_i * IDF_i = \left(t_{i1} * \log \frac{N}{n_1}, \dots, t_{ij} * \log \frac{N}{n_j}, \dots, t_{im} * \log \frac{N}{n_m} \right) \quad (1)$$

where m represents the total number of different keywords maintained in database 226, t_{ij} represents the frequency of keyword j appearing in the extracted set of

1 words associated with image i , n_j represents the number of images identified in
2 database 140 that contain the keyword j , and N represents the total number of
3 images in database 140. Each keyword in the text feature vector of an image is
4 thus weighted based on how frequently it appears in the text associated with the
5 image as well as how frequently it appears in the text associated with all images
6 identified in database 226. The resultant text feature vector D_i for image i thus
7 includes a numerical element for each word that is in the text associated with at
8 least one image identified in database 226 (if the word is not associated with
9 image i then the value for that element is zero).

10 Each time new high-level semantic text feature vectors are added to
11 database 228, the previously generated feature vectors are re-generated. crawler
12 modules 212 and 214 may generate (and re-generate) feature vectors based on the
13 features in database 226 as soon as new features are added to database 226, or
14 alternatively wait for multiple new features to be added to database 226, or wait
15 for a particular time (e.g., wait until early morning when fewer users will be
16 accessing a computer's resources).

17 Accordingly, the personal media database 226 is personal to a particular
18 user because it indexes all media objects that the particular user has accessed or
19 accumulated from the digital world, including media content from the Web, the
20 local machine 102, and all other media content stores 104 such as e-mail and other
21 composite documents. Once accumulated or otherwise accessed media content is
22 indexed by semantic text features, text-based search of the media files is possible.

23 For instance, U.S. patent application serial no. 09/805,626 to Li et al., filed
24 on March 13, 2001, titled "A Media Content Search Engine Incorporating Text
25 Content and User Log Mining", which is assigned to the assignee hereof and

hereby incorporated by reference, describes searching a database using semantic text features of media content.

The User Prediction Component

The prediction module 216 monitors a user's typing actions and guesses or anticipates whether the user may want to insert a media object based on the user intention model 232, which is described in greater detail below. To precisely predict the user's intention, the prediction module 216 generates the user intention model 232 based on a set of training data 236. For instance, the user's intention can be modeled using a Bayesian Belief Network (BBN) to represent probabilistic relationships among three levels of semantic features: lexicography or "lexics", syntax, and patterns. BBNs are known tools to represent probabilistic relationships. The user intention modeling process is presented in greater detail below in reference to the learning module 222.

The prediction module 216 uses the user intention model 232 and typed user text information to anticipate whether the user may want to insert a media file, and if so, the type of media file to insert. Specifically, the prediction module 216 extracts a set of keyword features from the text that the user has just typed and inputs the extracted keyword features to the BBN. The probabilities of all predefined user intentions are calculated based on the input keyword features and the one with the largest magnitude is chosen as the predicted user intention.

The prediction module 216 may determine or predict that a user desires to use/insert a media file into a document based on what a user types. This information can be used to predict even the context of the media file(s) that the user may wish to access. For instance, when the user is writing a document (e.g., an email), after the user types in text such as "The following are some pictures of

1 digital cassette recorder”, the prediction module 216 analyzes the text to guess that
2 the user may want to insert some pictures of a digital cassette recorder, and
3 therefore automatically activate the media search engine 218, which is discussed
4 in greater detail below, to locate media content that corresponds to digital cassette
5 recorders.

6 Fig. 8 shows an exemplary user interface 800 to present media suggestions
7 (e.g., filenames) for a user to insert into a document 802 based on what a user has
8 typed 804 into a window (e.g., an e-mail message). In this example, the user has
9 typed text 804 into an e-mail application window 802. The text 804 indicates that
10 “Attached are some pictures of ‘model name’ VCR”. The prediction module 216
11 analyzes this text 804 to guess that the user may want to “attach” some “pictures”
12 of a video cassette recorder (i.e., “VCR”) into the e-mail message 802.
13 Responsive to this guess, the media search engine 218 (discussed below) is
14 activated to locate media content that corresponds to VCRs. Upon locating such
15 corresponding media this information is presented (e.g., by the suggestion module
16 220—which is discussed in greater detail below) to the in a window 800 (e.g., a
17 media player window).

18 The media player window 800, in this example, includes an area 806 to
19 indicate that the search result has been incorporated into the media library or
20 personal media database 226. Window 808 indicates suggested media content
21 (e.g., filenames) based on the user input text 804. The user can simply select and
22 drag and drop the suggested media content 808 into the document 802 if one or
23 more of the suggestions are correct. The suggestions 808 can also be edited to
24 more closely approximate or indicate desired content—or simply ignored by the
25 user (e.g., a cancel button on a dialog box can be selected).

1 Windows 800, and 806 - 812 represent only an example of a user interface
2 with which to present suggested media content to a user based on the media
3 agent's 210 determination that a user desires to insert media content into a
4 document. For instance, there are a number of different ways for the media agent
5 to detect the user's desire to insert media content. The user may select a drop-
6 down menu item to indicate that an image is to be inserted at the current location
7 in the document, information in text surrounding an insert point (e.g., "see the
8 following diagram") may indicate that media content is to be inserted, and so on.

9 More details of the prediction process 216 are presented below in reference
10 to the user intention model 232.

11 The Search Engine Component

12 If it is determined that the user wants to access media content (e.g., to insert
13 something into a composite document), the media agent 210 uses the media search
14 engine 218 to locate relevant media objects based either on a search query that is
15 explicitly specified by the user or automatically guessed by the prediction module
16 216.

17 A user generates a search query by inputting a textual description of the
18 search criteria pertaining to the types of media content desired. The textual
19 description is then converted to a text feature vector and stored as a query vector
20 234; otherwise the prediction module 216 has automatically generated the query
21 vector 234 responsive to user actions (e.g., typing text).

22 A query vector 234 is generated by extracting keywords from search
23 criteria (e.g., user input) and building the query vector (having the same number of
24 elements as the semantic text feature vectors in database 226, and each element
25 corresponding to the same keyword as the corresponding element in the text

1 feature vectors) by assigning a value of one to the element corresponding to each
2 extracted keyword and a value of zero for the other elements. If an image is used
3 for the search criteria, then keywords of any text description corresponding to that
4 image are extracted and used to generate the initial high-level query vector. The
5 keywords can be extracted in the same manner as discussed above with reference
6 to online and offline crawler modules 212 and 214.

7 The high-level query vector 234 is then generated by assigning a value of
8 one to the element corresponding to each extracted keyword and a value of zero
9 for all other elements. If the image retrieval process is initiated based on both an
10 input text description and an input image, the high-level query vector is generated
11 based on extracted keywords from both the input text and the input image. For
12 example, initial vectors may be generated as discussed above (assigning a value of
13 one to the element corresponding to each keyword), and then the vectors
14 combined (e.g., elements added together or averaged on a per-element basis) to
15 generate the initial high-level query vector 234.

16 The search engine 218 uses a matching algorithm to determine the most
17 relevant media objects that match to the user's intent represented by the generated
18 query vector 234. The matching algorithm calculates semantic similarity between
19 the query vector 234 and each media object represented in the personal media
20 database 226. Semantic similarity is calculated using a dot product of the query's
21 semantic feature vector 234 and the media object's semantic feature vector.

22 For instance, the similarity, referred to as $S_h(q_h, D_i)$, between the high-
23 level query vector q_h and the high-level feature vector of the image D_i , referred to
24 as D_h , is calculated using the dot product of the query's text feature vector and the
25 image's text feature vector as follows, which is a normalized similarity.

$$S_h(q_h, D_{i_h}) = \frac{q_h \bullet D_{i_h}}{|q_h| |D_{i_h}|}.$$

The Suggestion Component

Once the search engine finds a set of relevant media objects in the personal media database 226, the suggestion module 220 shows (e.g., via the display 238) search engine 218 results to the user in a sorted list (e.g., in a dialog box) according to their semantic similarity to the query vector 234. Each object is displayed with a short paragraph of text or a few keywords to describe its content. The user may select an item from the list for acceptance. For instance, the user may select a suggested item by double clicking it or by dragging and dropping one or more of the suggested items from the display into a document.

Additionally, if the user places a cursor over a suggested item such as a suggested media content item or filename, the suggestion module 220 may display all or a portion (e.g., keywords) of the semantic text stored in the personal media database 226 that corresponds to the suggested item. This additional information can be displayed in a number of different ways such as in a hovering window near the cursor hot-point, in a status bar, and so on. Moreover, the user may decide at this point to modify the semantic descriptions of the media objects in the database 226 to more particularly indicate the semantics of the media content.

Additionally, when a user wants to save or download a media object (e.g., a multimedia file, html file, audio file, video file, image file, and so on) from a media source such as from the Web or from an e-mail message, the suggestion module 220 can include a "save-as" advisor 221 to present one or more suggested filenames for the user to utilize to save or download the media object to the personalized media database 226. These filenames are presented in a "Save-As"

1 dialog box or window on the display 238 and are based on semantic features that
2 are extracted from the media object and/or the media source. Such semantic
3 features include, for example, filenames, surrounding text, page titles, hyperlinks,
4 and so on. These semantic features are extracted from the media source by the on-
5 line crawler 212.

6 For instance the "save as" advisor 221 is activated when a user wants to
7 save or download a media object from the Web (i.e., a Web page). The "Save As"
8 Advisor automatically collects and extracts semantic information such as one or
9 more keywords from the Web page. From these extracted keywords, the advisor
10 suggests a list of corresponding filenames for a user to select. The user can
11 modify or choose a suggested filename to use as the filename of the saved media
12 object on the local machine 102.

13 The Learning Component

14 Learning is a significant aspect of the media agent 210. The media agent
15 210 improves performance based on relevance feedback from user interactions
16 with the system. The users' interactions are recorded in the user action log 228.
17 The self-learning mechanism of the media agent 210 is implemented in a number
18 of aspects, including: (a) learning to refine semantic features of accumulated
19 media files; (b) learning user preferences models for automatically indexing non-
20 visited but relevant media files; and (c) learning the user intention model 232 to
21 provide more accurate suggestions to a user.

22 Responsive to user selection of one or more of the suggestion module 220
23 displayed suggestions, the learning module 222 automatically refines the semantic
24 features of the search query 234 and updates the semantic indexing of the media
25 objects in the personal media database 226. To accomplish this, the learning

1 module 222 accesses relevance feedback from the user log 228 and updates the
2 query vectors to reflect the relevance feedback provided by the user. The query
3 vector 234 is modified as follows:

$$4 \quad Q' = Q + \beta \frac{\sum Q^+}{n^+} - \gamma \frac{\sum Q^-}{n^-}$$

5 where Q' represents the updated query vector, Q represents the original query
6 vector, Q^+ represents the set of feature vectors of user selected media content, n^+
7 represents the number of user selected media content, Q^- represents the set of
8 feature vectors of the non-selected media content, n^- represents the number of
9 non-selected media content, β represents a weighting for positive feedback, and
10 γ represents a weighting for negative feedback. Initially, the values of β and γ
11 are set empirically, such as $\beta = 1.0$ and $\gamma = 0.5$. Alternatively, if some training
12 data is available, the parameters can be tuned using the training data to improve
13 the performance of the retrieval.

14 If a query vector 234 did not previously exist, then an initial query vector
15 234 is generated based on the relevance feedback. For example, feature vectors of
16 the relevant images may be averaged together to generate a corresponding
17 semantic text query vector to store in the personal media database 226.

18 In this manner, suggested semantic features that result in positive user
19 feedback are reinforced in the personal media database 226. Additionally, by
20 learning from the user's log 228 of whether the user accepts or rejects suggestions,
21 the media agent 210 determines appropriate times to provide suggestions,
22 potentially saving processing time (e.g., searches). Additionally, user habits can
23 be determined to anticipate when media content suggestions (i.e., provided by the
24 suggestion module 220) may or may not be desired. Additionally, frequently

1 accessed media files usually show the user's preferences and profiles, which can
2 be learned more precisely by recording user actions over a period of time. Once a
3 user preference model 230 is known, the media agent 210 (i.e., the online or
4 offline crawlers 212 and 214) may automatically collect media objects pertaining
5 to the user's interests from various media content sources 104.

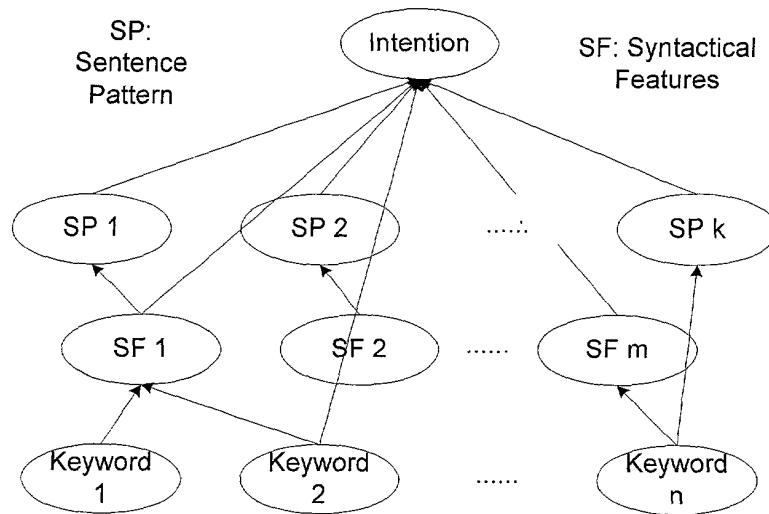
6 The User Intention Modeling Component

7 The self-learning mechanism 222 also includes user intention modeling 232
8 and preference modeling 230 based on the log 228 of a user's actions on
9 accumulated media content. Many kinds of user activities, including mouse
10 movement and typing can be used to learn and predict the user's intentions. For
11 instance, when the user is writing a new e-mail and has typed "Here is an
12 interesting picture download from the web", the probability of the user's intention
13 of inserting an image into the e-mail body as an attachment is very high.
14 Therefore, the media agent 210 (i.e., the prediction module 216) can predict that
15 user wants to insert an image in the e-mail. If the user's intention is to insert, the
16 suggestion module 220 can provide potential images for the user to insert based on
17 other text information the user has typed or will type.

18 All text factors that may imply the user's intentions are referred to as
19 linguistic features. A Bayesian Belief Network is used to precisely represent the
20 dependencies and probabilities among the linguistic features and the user's
21 intentions. Three levels of linguistic features are defined: lexics, syntax, and a
22 partial or completely instantiated sentence pattern. A lexical feature is a single
23 word extracted from the text. A syntactical feature is the syntax structure of a
24 sentence. An instantiated pattern feature is a frequently used sentence structure
25 with some of its syntactical units instantiated with certain words or phrases, e.g.,

“Here it is a...” and “Attached please find ...” The particular Bayesian Belief Network used to represent the user’s intention model 232 is illustrated below in table 1.

TABLE 1
Example of User Intention Modeling using a Bayesian Belief Network



Initially, the user intention model of Table 1 is empty but is subsequently learned using the user’s log 228 (a set of user action records) as training data. The user’s log 228 records user action records to train the intention model. Each user action record contains a text part and a tag of whether a media file is attached. For instance, an e-mail message could have a text body and a media file attachment. The text part is parsed such that all words (lexical features) are extracted from the sentences and are stemmed.

At the lexical level, direct association between keyword features and user intentions is determined through training. A fast algorithm proposed by Agrawal et al. [1] can be used to generate rules for determining associations between

keywords and intention(s). The rules represent the causality relationship between keywords and intentions.

For example, a set of rules identify whether there is a causal relationship between certain keywords and the intention to insert a media file. The causality rules are further constrained by two parameters: α (*Support of Item Sets*) and β (*Confidence of Association Rule*). The first parameter (α), which depicts a scope that the rules can be applied to, is expressed by the percentage of those records that contain the same keyword as evidence. The second parameter (β) depicts the probability that the rule stands, i.e., the probability of the intention given the appearance of the keyword. The generated rules are evaluated based on the values of these two parameters. The higher the two values, the better the rules. Those rules with parameters higher than certain thresholds (e.g., $\alpha = 0.03, \beta = 0.6$) are selected to build the Bayesian Belief Network.

The Intention Prediction Process

Once direct associations between keyword features and user intentions are determined through training, the intention model 232 for a user can be used by the prediction module 216 to predict the user's intention based on what the user has just typed.

To accomplish this, a set of keyword features represented by $\langle a_1, a_2 \dots a_n \rangle$ are extracted from the text typed by user. The prediction module 216 then calculates the probabilities of all predefined user intentions (V), and selects the intention with the biggest probability (v_{map}) using the following equation [11].

$$\begin{aligned} v_{map} &= \arg \max_{v_j \in V} P(v_j | a_1, a_2, \dots, a_n) \\ &= \arg \max_{v_j \in V} P(a_1, a_2, \dots, a_n | v_j) P(v_j) \end{aligned} \quad (2)$$

where $P(a_1, a_2, \dots, a_n | v_j) = \prod_{i=1}^n P(a_i | Parents(a_i), v_j)$.

1 In addition to lexical features, other informative features are used to
2 precisely predict the user's intentions. For instance, natural language processing
3 (NLP) technologies can be utilized to analyze sentence structures of the text. NLP
4 can analyze a sentence and parse it into a tree structure. The highest-level
5 sentence structures are utilized.

6 For example, "Here are some photos" is parsed into the following sentence
7 structure: AVP ("here"), VERB ("be"), NP ("some photos"), wherein "AVP"
8 represents an indication of an adverb phrase element, and "NP" represents a noun
9 phrase element. Such syntactical features are used to determine additional useful
10 information. This method [1] can also be used to generate association rules
11 between these syntactical features and user intentions.

12 Use of syntactical features improves user intention prediction precision.
13 However, certain sentence patterns, such as, "here is something" in an e-mail
14 message typically indicates that a user intends to insert an attachment. The
15 sentence structure is AVP+VERB+NP. Yet, the sentence "how are you" has the
16 same structure and indicates a different intention. Therefore, parts of the sentence
17 structure are further evaluated to locate instantiated patterns that may strongly
18 indicate user's intentions. An instantiated pattern is a sentence structure specified
19 with a pattern of words.

20 The rules generated at the lexical level using the method of [1] are
21 substantially specific and lack complete representation of user intent. Hence,
22 association rules are generated for instantiated pattern features based on the
23 association rules found between syntactical features and user intentions. By
24 instantiating parts of the syntactical features with certain frequently used words or
25 phrases, association rules are generated at the instantiation pattern level, which is

1 more general than the syntax level rules and more specific than the lexics level
2 rules.

3 Since each syntactical unit can be replaced by many words, all
4 combinations of words found in the training data and syntactical units are tested in
5 a breadth-first order. Only those instantiated patterns that have α and β
6 parameters (of the association rules) that are greater than certain thresholds are
7 selected for user intention prediction.

8 The User Preferences Modeling Component

9 A user model includes many things about a user. The association rules and
10 the intention models discussed above are part of the user intention model 232.
11 This section focuses on how to identify user interests and preferences (e.g., the
12 user preference model 230) from the user's interaction history with the media
13 agent 210 as identified in the user action log 228.

14 User preferences are modeled by analyzing semantic features of the media
15 files that the user has accumulated and frequently used. By doing so, a scope of
16 media contents matching user interest is identified. Once user preferences models
17 230 are identified, the media agent 210 provides appropriate suggestions (e.g., via
18 the suggestion module 220) or preferred media files automatically collected from
19 all possible sources by the offline crawler 214. Additionally, media files on the
20 local machine 102 can be automatically and periodically sought for better
21 indexing, clustering, and/or classification. Moreover, media files can be shared
22 with other users that have similar user preferences models 230.

23 To improve user preference modeling, several different preference models,
24 each of which can be represented by a list of keywords, can be maintained for a
25 user. For instance, all user log records 228 are clustered into several preferences

1 clusters based on their semantic similarity. The semantic similarity for two
2 keyword vectors is calculated using their dot product and normalized through the
3 cosine method [5][13] (these methods were discussed above in reference to
4 determining semantic similarity with respect to search engine 218 results). Each
5 cluster corresponds to a preference model 230 for the user, which is represented by
6 a keyword frequency vector formed by the top 10 frequently used keywords
7 (except for stop words) and their frequency in the user log cluster. A user
8 preference model is therefore represented by $m = \langle k_1, k_2, \dots, k_{10} \rangle$.

9 Whether a media file or object is of interest to the user also depends on the
10 semantic similarity between the media object and one of the user preferences
11 models 230. The one with the largest similarity value (which is also large enough,
12 e.g., larger than a threshold) is considered as relevant to the user's interest.
13 Similarities between two user preference models 230 are compared by calculating
14 the dot product of their keyword frequency vectors.

15 Modeling user's preferences based on keyword probability is another
16 approach to determining keyword frequency in text documents. Specifically, the
17 Naïve Bayes approach is used with respect to all words and their probabilities to
18 form a keyword probability vector to model a preference [11]. The probability of
19 word w_k is estimated using the following equation:

$$20 \quad P(w_k | m_j) = \frac{n_k + 1}{n + |Vocabulary|}, \quad (3)$$

21 where n is the total number of words (or actually, the total length of text)
22 existing within the training data, which are all user log records in the cluster
23 corresponding to the user preference model m_j , n_k is the number of times that word
24 w_k is found among these n words, and $|Vocabulary|$ is the total number of distinct

words found in the training data. In comparison, equation (2) is simply the term frequency combined with a smoothing function.

Given a multimedia document D represented by $\langle w_1, w_2, \dots, w_n \rangle$, the most probable user preference model M_{NB} is calculated using the Naïve Bayes approach as follows.

$$\begin{aligned}
 m_{NB} &= \operatorname{argmax}_{m_j \in M} P(m_j | w_1, w_2 \dots w_n) \\
 &= \operatorname{argmax}_{m_j \in M} P(w_1, w_2 \dots w_n | m_j) P(m_j) \\
 &= \operatorname{argmax}_{m_j \in M} P(m_j) \prod_k P(w_k | m_j)
 \end{aligned} \tag{4}$$

P_{m_j} is the prior of m_j , which can be considered as of a uniform distribution initially. The approach assumes that the probability of a word is independent of others or its position within the text. Note that this assumption is not always true. However, in practice, the Naïve Bayesian learner performs remarkably well in many text classification problems despite this independence assumption [11].

$P(m_j | w_1, w_2 \dots w_n)$ is comparable among different m_j , and can therefore be used to find a better model. However, $P(m_j | w_1, w_2 \dots w_n)$ is not comparable among different D , since it differs in magnitude for different lengths of keyword vectors. To judge whether D is of the user's interest, another metric is required that is comparable among different D such that a value larger than a threshold means that the document is of the user's preference. First at all, due to multiple multiplications in equation (3), a geometrical mean is considered in normalizing as follows.

$$\log(\sqrt[n_w]{P(m_{NB}) \prod_i P(w_i | m_j)}) = \frac{\log(P(m_{NB} | w_1, w_2 \dots w_n))}{n_w},$$

where n_w is the number of distinct keywords in document D matched with keywords in the model M_{NB} . Secondly, a factor of matched keyword percentage is considered such that the document containing a larger percentage of keywords in

1 D matched in the model will get a higher metric value and is therefore more
2 relevant to the user's preference model. Hence,

$$\frac{n_D}{n_u}$$

3
4 is multiplied, where n_D is the total number of words in D . Finally, the
5 metric is defined as follows, which is used to measure the relevancy of D to the
6 user's preference model.

$$P_{norm}(D) = \frac{n_D * \log(P(m_{NB} | w_1, w_2, \dots, w_n))}{n_v^2}, \quad (5)$$

7
8 Using the Bayesian method, the similarity between two user preference
9 models is calculated. One of the metrics follows:

$$Sim(m_1, m_2) = \frac{P(m_1 | m_2) + P(m_2 | m_1)}{2}, \quad (6)$$

10
11 where, m_1 and m_2 are two user preference models represented by two keyword
12 vectors.

13 The Semantic Feature Refinement Component

14 It is not initially known whether a text paragraph in a document is relevant
15 to a media object in the same document. In light of this, the media agent 210 may
16 save many features surrounding the media object. These saved or extracted
17 feature indices stored in the personal media database 226 may contain many
18 redundant or non-related texts. Redundant text features decrease performance
19 searches on the database 226. Additionally, providing such redundant or non-
20 related text in a search result may confuse a user. Hence the database 226 text
21 features are refined either by deleting non-relevant features or by decreasing their
22 importance or relevance weights. These aims are accomplished in view of the user
23 action log 228.

1 Each record in the user log 228 includes a user inserted text portion (e.g.,
2 user typed, spoken, or otherwise inserted), and an attached media object. In light
3 of this, all text portions of the user log records 228 that contain the same media
4 objects are combined together. Keyword frequencies across this combined text are
5 calculated. These keyword frequency calculations replace the keyword weights in
6 the original semantic features that correspond to the same media object.

7 Moreover, a number of top frequency keywords (e.g., the three (3) top
8 frequency keywords) are selected along with their respective locations relative to
9 the media object in the original document layout. Other media objects that came
10 from similar sources (e.g., the same web page, the same website, and e-mail from
11 the same person) are then examined. If a keyword in the semantic features of the
12 other evaluated media came from the same relative locations of those top
13 frequency keywords, there is confidence that the keyword's weight should be
14 increased some amount (e.g., 10%). In this way, portions of original document
15 layout that are more relevant to the semantics of the media object are determined.

16 **An Exemplary Media Agent Process and Data Flow**

17 Fig. 3 shows exemplary aspects of process and data flows between modules
18 and data sinks in the media agent module 210. Specifically, Fig. 3 shows
19 sequences in which data transfer, use, and transformation are performed during the
20 execution of the media agent 210 of Fig. 2. Data flow is illustrated with lines
21 between respective modules and data sinks. Actions or events that trigger or cause
22 execution and subsequent data flow are shown with lines capped with circles
23 rather than arrows.

24 Sources of media content 300 include, for example, the WWW or the
25 Web 302, e-mail messages 304, local and remote documents or Web pages 306,

1 local and remote file folders 308, and so on. These media content sources 300 are
2 only a few examples of the many possible sources of the media content pieces 106
3 of Fig. 1. On-line and offline crawler modules 212 and 214 retrieve semantic text
4 description from these media content sources 300 to subsequently store into the
5 personal media database 226. Trigger 310 shows that the on-line crawler is
6 activated by user actions 312 such as accessing the Web 302, e-mails 304,
7 documents or Web pages 306, file folders 308, user "save-as" actions 316,
8 browsing 318, text insertion 320, media insertion 321, and so on.

9 The offline crawler 214 is activated to access any number of these media
10 sources 300 at system idle or as otherwise indicated by a user. The particular
11 media sources 300 that are to be accessed by the offline crawler 214 are
12 determined by information stored in the user preference models 230. As discussed
13 above, the learning module 222 and more specifically, the user preferences
14 modeling sub-module generates the user preference models 230 from records of
15 user actions stored in the user actions log 228. These data flows from the user
16 action 312, to the user action log 228, to the learning module 222, the user
17 preference models 230, and the offline crawler module 214 are shown by lines
18 314-1 through 314-4. Note that the dataflow 314-4 is a trigger. This means that a
19 user can implicitly or explicitly express a preference for when and where the
20 offline crawler 214 is to obtain its information.

21 The user intention model 232 stores information that is used by the
22 prediction module 216 to predict or anticipate user intention. As discussed above,
23 the user intention model data store 232 is generated by learning module 222 and
24 more specifically by user intention modeling sub-module based on lexics, syntax,
25 and/or patterns evaluated in training data such as data obtained from user action

1 log 228. The user action log 228 stores records of the user's actions. Each record
2 includes a text portion and an indication of whether or not a media file is part of
3 the action. Direct associations between the user actions, corresponding media
4 content, and user intentions are determined on lexical, syntactical, and/or pattern
5 basis. These direct associations are stored in the user intention model 232 data
6 store. Arrows 314-1, 323-1, and 323-2 represent the data flow from the user
7 action 312 to the user intention model data store 232.

8 Certain user actions 312 cause media agent 210 modules to predict or
9 anticipate user actions to provide semantically related suggestions to the user.
10 Examples of such user actions 312 include, a "save as..." action 316, a "browse
11 ..." action 318, the insertion of text 320, and an insert item action 322. This
12 action based trigger is illustrated by line 326-1. Responsive to such user actions,
13 prediction module 216 evaluates direct associations between the user actions,
14 corresponding media content, and user intentions. (These direct associations are
15 stored in the user intention model data store 232 as shown by dataflow line 323-3).
16 The prediction module 216 anticipates that the user desires to work with media
17 files based on the user action 312 (e.g., the text information typed by the user, the
18 selection of an insert media content menu item, and so on).

19 If the prediction module 216 determines that the user desires to work with
20 media files, the prediction module 216 generates a potential search query vector
21 (e.g., the query vector 234 of Fig. 2) from relevant information derived from
22 evaluation of the user action 312 in view of the information in the user intention
23 model 232. This query vector may have been partially or wholly formulated from
24 text typed in by the user for any number of reasons, including in response to an
25 explicit user search for information. The prediction module 216 triggers and

1 communicates the predicted query vector to the search engine 218. This particular
2 data flow is represented by arrows 326-2, through 326-3. Note that line 326-1
3 shows that user action 312 triggered the prediction module 216. Note also that
4 line 326-2 shows that the prediction module 216 triggered the search engine 218.

5 The search engine 218 receives a search query vector (e.g., the query vector
6 234 of Fig. 2) that may have been wholly or partially generated from information
7 (e.g., text) input by the user or wholly or partially generated by the prediction
8 module 216. The search engine 218 evaluates this query in view of the semantic
9 media content text indices stored in the personal media database 226. (Recall that
10 these indices are created by the online and offline crawlers 212 and 214. These
11 indices are also created by the feature refinement sub-module of the learning
12 module 222 as described below).

13 If a relevant set of media objects in the personal media database 226 are
14 identified, the search engine 218 triggers and communicates the identified media
15 objects to the suggestion module 220. This information may be sorted and
16 communicated to the suggestion module 220 according to each items semantic
17 similarity to the search query. These data flow are represented by lines 326-1
18 through 326-4. Note that lines 326-1, 326-2, and 326-4 are triggers.

19 The suggestion module 220 receives a list of relevant media content from
20 the search engine 218. This information is displayed to a user for viewing and
21 response (e.g., selection or editing of a suggestion, editing of the semantic text
22 corresponding to the suggestion, cancellation of the task, and so on). These data
23 flow are represented by lines 326-4 through 326-5. Note that line 326-4 is a
24 trigger.

1 The learning module 222 and specifically the feature refinement sub-
2 module, refines the semantic media content text indices stored in the personal
3 media database 226. To accomplish this, the feature refinement sub-module
4 evaluates text and corresponding media content in the user action log 228 to
5 evaluate corresponding keyword frequencies or keyword relevance. The feature
6 refinement sub-module uses this keyword evaluation to redefine keyword weights
7 in the personal media database 226 to correlate with a compiled history of user
8 actions. This redefinition of keyword weights may result in removal of certain
9 226 keywords in indices in the database that are determined to be semantically
10 non-related at that point in time. These data flow are represented by lines 328-1
11 and 328-2.

12 **An Exemplary Media Agent Procedure**

13 Fig. 4 shows an exemplary procedure 400 to automatically collect, manage,
14 and suggest information corresponding to personalized use of media content.
15 More specifically, Fig. 4 shows a procedure 400 for a media agent 210 of Figs. 2
16 and 3 to determine whether offline gathering of media content semantics, online
17 gathering of media content semantics, preference and intention modeling, or user
18 intention prediction and suggestion procedures should be performed.

19 At block 402, the procedure determines if a user action (e.g., a mouse
20 move, a key press, saving of a file, downloading a file, following a link or
21 hyperlink on a network, and so on) is received. If not, the procedure continues at
22 block 404, wherein it is determined if the system 102 of the media agent 210 is
23 idle for some reason (e.g., a pause between keystrokes, etc.). A system is in an
24 idle state when it is operational and in service but one or more processing cycles is

1 still available for use. Having determined that the system is not in an idle state,
2 the procedure continues at block 402, as described above.

3 If the system is idle (block 404), the procedure 400 continues at block 406,
4 wherein the offline crawler program module 214 extracts semantic text features
5 from media content sources (e.g., e-mails, documents, memory caches, etc.), if
6 any, according to the user preference model 230. The user preference model 230
7 indicates learned aspects of a user's behavior with respect to media content
8 locations, preferred or frequently accessed media content, and so on. These
9 learned aspects identify preferred media content and respective semantic features
10 to extract and store while the system is idle.

11 At block 408, the procedure stores any extracted semantic features and
12 corresponding media content (block 404) into the user's personalized media
13 database 226. In this manner the database 226 includes personalized semantic
14 indices (PSI) that reflect all of the media content accessed by the user. Since each
15 user may have different preferences of his/her favorite media objects, the personal
16 index may differ from user to user. The procedure continues at block 402.

17 At block 402, responsive to receiving a user action, the procedure continues
18 at block 410, wherein a user action log (i.e., log 228 of Figs. 2 and 3) is updated to
19 include a record of the text and/or media content corresponding to the user's
20 action (block 402). At block 412, it is determined if the action corresponds to user
21 access of a URL, opening of a media file, or downloading a file (e.g., saving a
22 file). If so, the procedure continues at on-page reference "B", as shown in Fig. 5.

23 Fig. 5 shows further aspects of an exemplary procedure 400 to
24 automatically collect, manage, and suggest information corresponding to
25 personalized use of media content. More specifically, Fig. 5 shows further aspects

1 of a procedure for a media agent 210 of Figs. 2 and 3 to perform online gathering
2 of media content semantics and preference and intention modeling. Reference "B"
3 indicates that procedure 400 executes blocks 502 through 510. Although the
4 blocks are orderly numbered, the ordering does not imply any preferred sequence
5 of execution. For instance, blocks 502 and 504 may be executed before blocks
6 506 through 510, vice versa, and so on.

7 At block 502, the procedure 400 (i.e., the online crawler 212 of Figs. 2
8 and 3) extracts semantic media content features (i.e., text features) from the media
9 content itself and/or from a document (e.g., e-mail, Web page, etc.) corresponding
10 to the media content. Recall that this operation (block 502) is performed
11 responsive to a user action (e.g., a URL access, an open file action, a save as
12 action, and so on). At block 504, the extracted semantic features and
13 corresponding media content are stored in the user's personal media database 226.
14 It can be appreciated that the semantic features can be stored separately, if desired,
15 from the media content.

16 At block 506, user preference modeling is performed. As discussed above,
17 the learning module 222 of Figs. 2 and 3 and more specifically, the user
18 preferences modeling sub-module (see, Fig. 3) generates the user preference
19 models 230 from records of user actions stored in the user actions log 228.

20 At block 508, the procedure 400 performs user intention modeling to store
21 information that is used by the prediction module 216 of Figs. 2 and 3 to predict or
22 anticipate user intention. As discussed above, the user intention model data store
23 232 is generated by learning module 222 and more specifically by user intention
24 modeling sub-module of Fig. 3 based on lexics, syntax, and/or patterns evaluated
25 in training data such as data obtained from user action log 228.

1 At block 510, the procedure 400 refines the semantic features
2 corresponding to media content stored in the personal media database 226 of Figs.
3 2 and 3. The feature refinement sub-module of Fig. 3 performs this operation by
4 evaluating text features and corresponding media content in the user action log
5 228 to evaluate corresponding keyword frequencies or keyword relevance. The
6 feature refinement sub-module uses this keyword evaluation to redefine or update
7 keyword weights in the personal media database 226 to correlate or with a
8 compiled history of user actions. At this point, the procedure 400 continues at
9 block 402, as shown by the on-page reference "A" of Fig. 4.

10 Recall that at block 412 of Fig. 4, the procedure 400 determines if the
11 identified user action (block 402) is of a particular type of user action (e.g., URL
12 access, media file open, media file save as, and so on). If so, the discussed
13 procedures of Fig. 5 were performed. However, if the user action was not of the
14 particular type, the procedure 400 continues at block 602 of Fig. 6, as illustrated
15 by on-page reference "C".

16 Fig. 6 shows further aspects of exemplary procedures to automatically
17 collect, manage, and suggest information corresponding to personalized use of
18 media content. More specifically, Fig. 6 shows further aspects of a procedure for
19 a media agent of Figs. 2 and 3 to determine whether preference and intention
20 modeling or user intention prediction and suggestion procedures should be
21 performed. At block 602, the procedure 400 determines if the user action (block
22 402) is an explicit user search for a media object, an object insertion action, or
23 action corresponding to a document edit (e.g., e-mail, a word-processing
24 document, etc.). If not, the procedure continues at block 402, as indicated by the
25 on-page reference "A" of page 4.

1 Otherwise, at block 604, the procedure (i.e., the prediction module 216 of
2 Figs. 2 and 3) generates a set of media content predictions using a user intention
3 model 232 of Figs. 2 and 3. A search query vector (e.g., the query vector 234 of
4 Fig. 2) is generated from the media content predictions in view of the user action
5 (block 602). At block 606, the procedure 400 uses the generated query vector
6 (block 604) to search the user's personal media database 226 of Figs. 2 and 3 for
7 corresponding media content. At block 608, identified media content information
8 (e.g., file names, URLs, etc...) is displayed or "suggested" to the user for
9 subsequent evaluation, selection, and/or other response (e.g., editing). The
10 procedure continues at block 402, as indicated by the on-page reference "A" of
11 page 4.

12 **An Exemplary Suitable Computing Environment**

13 Fig. 7 illustrates aspects of an exemplary suitable operating environment in
14 which a media agent to semantically index, suggest, and retrieve media content
15 information according to personal usage patterns may be implemented. The
16 illustrated operating environment is only one example of a suitable operating
17 environment and is not intended to suggest any limitation as to the scope of use or
18 functionality of the invention. Other well known computing systems,
19 environments, and/or configurations that may be suitable for use with the
20 invention include, but are not limited to, personal computers, server computers,
21 hand-held or laptop devices, multiprocessor systems, microprocessor-based
22 systems, programmable consumer electronics (e.g., digital video recorders),
23 gaming consoles, cellular telephones, network PCs, minicomputers, mainframe
24 computers, distributed computing environments that include any of the above
25 systems or devices, and the like.

1 Fig. 7 shows a general example of a computer 742 that can be used in
2 accordance with the described arrangements and procedures. Computer 742 is
3 shown as an example of a computer in which various embodiments of the
4 invention can be practiced, and can be used to implement, for example, a
5 client 102 of Fig. 1, a media agent 210, online and offline crawler components 212
6 and 214, prediction component 216, search engine component 218, suggestion
7 component 220, or a learning component 222 of Figs. 2 and 3, and so on.
8 Computer 742 includes one or more processors or processing units 744, a system
9 memory 746, and a bus 748 that couples various system components including the
10 system memory 746 to processors 744.

11 The bus 748 represents one or more of any of several types of bus
12 structures, including a memory bus or memory controller, a peripheral bus, an
13 accelerated graphics port, and a processor or local bus using any of a variety of
14 bus architectures. The system memory 746 includes read only memory
15 (ROM) 750 and random access memory (RAM) 752. A basic input/output system
16 (BIOS) 754, containing the basic routines that help to transfer information
17 between elements within computer 742, such as during start-up, is stored in
18 ROM 750. Computer 742 further includes a hard disk drive 756 for reading from
19 and writing to a hard disk, not shown, connected to bus 748 via a hard disk drive
20 interface 757 (e.g., a SCSI, ATA, or other type of interface); a magnetic disk
21 drive 758 for reading from and writing to a removable magnetic disk 760,
22 connected to bus 748 via a magnetic disk drive interface 761; and an optical disk
23 drive 762 for reading from and/or writing to a removable optical disk 764 such as
24 a CD ROM, DVD, or other optical media, connected to bus 748 via an optical
25 drive interface 765. The drives and their associated computer-readable media

1 provide nonvolatile storage of computer readable instructions, data structures,
2 program modules and other data for computer 742. Although the exemplary
3 environment described herein employs a hard disk, a removable magnetic disk 760
4 and a removable optical disk 764, it will be appreciated by those skilled in the art
5 that other types of computer readable media which can store data that is accessible
6 by a computer, such as magnetic cassettes, flash memory cards, random access
7 memories (RAMs), read only memories (ROM), and the like, may also be used in
8 the exemplary operating environment.

9 A number of program modules may be stored on the hard disk, magnetic
10 disk 760, optical disk 764, ROM 750, or RAM 752, including an operating
11 system 770, one or more application programs 772, other program modules 774,
12 and program data 776. A user may enter commands and information into
13 computer 742 through input devices such as keyboard 778 and pointing
14 device 780. Other input devices (not shown) may include a microphone, joystick,
15 game pad, satellite dish, scanner, or the like. These and other input devices are
16 connected to the processing unit 744 through an interface 768 that is coupled to
17 the system bus (e.g., a serial port interface, a parallel port interface, a universal
18 serial bus (USB) interface, etc.). A monitor 784 or other type of display device is
19 also connected to the system bus 748 via an interface, such as a video adapter 786.
20 In addition to the monitor, personal computers typically include other peripheral
21 output devices (not shown) such as speakers and printers.

22 Computer 742 operates in a networked environment using logical
23 connections to one or more remote computers, such as a remote computer 788.
24 The remote computer 788 may be another personal computer, a server, a router, a
25 network PC, a peer device or other common network node, and typically includes

1 many or all of the elements described above relative to computer 742, although
2 only a memory storage device 790 has been illustrated in Fig. 7. The logical
3 connections depicted in Fig. 7 include a local area network (LAN) 792 and a wide
4 area network (WAN) 794. Such networking environments are commonplace in
5 offices, enterprise-wide computer networks, intranets, and the Internet. In certain
6 embodiments of the invention, computer 742 executes an Internet Web browser
7 program (which may optionally be integrated into the operating system 770) such
8 as the "Internet Explorer" Web browser manufactured and distributed by Microsoft
9 Corporation of Redmond, Washington.

10 When used in a LAN networking environment, computer 742 is connected
11 to the local network 792 through a network interface or adapter 796. When used
12 in a WAN networking environment, computer 742 typically includes a modem 798
13 or other means for establishing communications over the wide area network 794,
14 such as the Internet. The modem 798, which may be internal or external, is
15 connected to the system bus 748 via a serial port interface 768. In a networked
16 environment, program modules depicted relative to the personal computer 742, or
17 portions thereof, may be stored in the remote memory storage device. It will be
18 appreciated that the network connections shown are exemplary and other means of
19 establishing a communications link between the computers may be used.

20 Computer 742 also includes a broadcast tuner 799. Broadcast tuner 799
21 receives broadcast signals either directly (e.g., analog or digital cable
22 transmissions fed directly into tuner 799) or via a reception device (e.g., via
23 antenna or satellite dish).

24 Computer 742 typically includes at least some form of computer readable
25 media. Computer readable media can be any available media that can be accessed

1 by computer 742. By way of example, and not limitation, computer readable
2 media may comprise computer storage media and communication media.
3 Computer storage media includes volatile and nonvolatile, removable and non-
4 removable media implemented in any method or technology for storage of
5 information such as computer readable instructions, data structures, program
6 modules or other data.

7 Computer storage media includes, but is not limited to, RAM, ROM,
8 EEPROM, flash memory or other memory technology, CD-ROM, digital versatile
9 disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic
10 disk storage or other magnetic storage devices, or any other media which can be
11 used to store the desired information and which can be accessed by computer 742.
12 Communication media typically embodies computer readable instructions, data
13 structures, program modules or other data in a modulated data signal such as a
14 carrier wave or other transport mechanism and includes any information delivery
15 media. The term "modulated data signal" means a signal that has one or more of
16 its characteristics set or changed in such a manner as to encode information in the
17 signal. By way of example, and not limitation, communication media includes
18 wired media such as wired network or direct-wired connection, and wireless media
19 such as acoustic, RF, infrared and other wireless media. Combinations of any of
20 the above should also be included within the scope of computer readable media.

21 The invention has been described in part in the general context of
22 computer-executable instructions, such as program modules, executed by one or
23 more computers or other devices. Generally, program modules include routines,
24 programs, objects, components, data structures, etc. that perform particular tasks
25 or implement particular abstract data types. Typically the functionality of the

1 program modules may be combined or distributed as desired in various
2 embodiments.

3 For purposes of illustration, programs and other executable program
4 components such as the operating system are illustrated herein as discrete blocks,
5 although it is recognized that such programs and components reside at various
6 times in different storage components of the computer, and are executed by the
7 data processor(s) of the computer.

8 Alternatively, the invention may be implemented in hardware or a
9 combination of hardware, software, and/or firmware. For example, one or more
10 application specific integrated circuits (ASICs) could be designed or programmed
11 to carry out the invention.

12 **Conclusion**

13 Although the description above uses language that is specific to structural
14 features and/or methodological acts, it is to be understood that the described
15 arrangements and procedures defined in the appended claims are not limited to the
16 specific features or acts described. Rather, the specific features and acts are
17 disclosed as exemplary forms of implementing the described arrangements and
18 procedures.